

## CS559 – Lecture 3

### Image Representation and Quantization



These are course notes (not used as slides)

Written by Mike Gleicher, Sept. 2005

© 2005 Michael L. Gleicher

## Last time



- How images are formed
- The eye as a sensor
- Today:
  - Properties of the eye as a sensor
  - What this means for graphics
  - How to represent images for display

## Types of photoreceptors: Rods



- Photopigment: Rhodopsin
  - Breaks into retinene + protein
  - Must be reassembled before can work again
- Very sensitive
  - Bright light means that it breaks down faster than it is regenerated
  - Less useful in bright light
- Blinded by bright light at night

## Cones



- Photopigments reform quickly
- Different types of cones sensitive to different kinds of light (color sensitivity)
  - Humans – 3 types of cones
    - Except for color blindness
  - Dogs – 1 type of cone
  - Many mammals (horses, cows, deer, ...) – 2 types
  - Ducks, Pigeons - 5 types (?)
    - Birds range in number – European Starling 4
    - <http://people.eku.edu/ritchison/birdbrain2.html>
- We'll talk about

## Persistence of Vision



- Photopigments take some time to regenerate
- If you see a flash, you sense it for a while afterwards
- This is NOT how you fuse movie frames together in order for it to seem continuous
  - This is actually hard psychological science that is not well understood
  - Integration happens as a higher level process in the brain
  - Many other effects

## “Flicker-based Displays”



- If something flashes fast enough, it seems to be continuous
  - Flicker frequency – approx 40-45 hz in a dim/dark room
  - Sensitivity varies with age and ambient brightness
- Used to create different types of displays
  - CRT
  - Movies

## How many megapixels is the eye?



- Density of photoreceptors varies (see book)
- Dense area of cones = **fovea**
  - Eye moves the scene around, fovea looks at a little piece and over time gets the whole picture
  - Saccade – movement of the eye to see different piece
  - Fixation –
- Wide angle view means “resolution” hard to talk about – easiest to talk about in terms of angle
- Discriminate about  $\frac{1}{2}$  minute of arc (for 20/20 vision)
  - At .5 meters, this is .1mm

## How sensitive is the eye?



- Amazing range!
  - Night vision – when eyes adjusted, camping
  - Bright daylight
    - Sunlight 10000.
    - Twilight 10.
    - Starlight 0.001
- Catch: at any given time, can't see this range
  - Adaptation – bright light, iris closes, lets in less light, ...
- At any given time, about 100:1 contrast ratio
  - This is a lot more than most displays
  - Better displays = more contrast
    - Often by blacker blacks

## High Dynamic Range Imagery



- Most sensors/displays have less range than eye
  - Certainly less range than scenes do
- What happens?
  - Bright areas – all white (no details)
  - Dark (shadow) areas – all black (no details)
- What to do?
  - Adjust exposure (time, aperture, sensitivity) to get the most important stuff
  - Acquire “High Dynamic Range” Imagery
    - Special sensors
    - Multiple exposures (at different settings) – cool thing to do
  - Tone Map -> display on device with less range
    - A chapter in the book we won't get to

## Perception of intensity



- Eye senses relative differences
  - Equivalent differences 50:100 20:40
    - Adaptation to current setting
- Can sense 1% differences
- At any given time 100:1 contrast ratio
- How many levels can you see in an image?
  - $1.01^{463} = 100.2$  (e.g. 463 1% differences = 100:1)
  - This is about 8 bits of precision (less than 9)
  - But its VERY non linear 1, 1.01, ..., 99.2, 100.2

## Non-linearity of intensity



- Non-linear mapping from “amount of light” to perceived brightness
- Want uniform mapping of intensities -> perception
  - Level 1, 2, 3, ..., 255 -> 1, 1.01, 1.02, ... 99, 100
- Worse: displays are non-linear too
  - Voltage -> amount of light is non-linear
  - Different displays are different
- Want to linearize the system
  - Intensity levels map nicely to perceived levels

## Gamma correction



- Idea: put a non-linear function between intensity and output
  - Done as the last step (usually) – after all computations
- Could create arbitrary functions for mapping
  - Too cumbersome
- Exponential is a good approximate model
  - Exponential non-linearity of perception
  - Exponential power laws in CRTs

## Modeling a display device



- 5/2 power law (five-halves)
  - Models physics of a CRT
  - Real CRTs are close, LCDs designed to be similar
- $L = M (i + \epsilon)^\gamma$ 
  - $i$  = input intensity value
  - $L$  = amount of light
  - $\epsilon$  = since zero isn't really black
  - $M$  = maximum intensity
  - $\gamma$  = specific property of display

## Linearizing the display



- Define a function  $g$  that corrects for non-linearity
- $L = M (g(i))^\gamma$  (ignoring  $\epsilon$ )
  - $G = 1/\gamma$
- Where do we get  $\gamma$  from?
  - Pick it so things look right
- Note: 1<sup>st</sup> order approximation (very simple)
  - Only 1 parameter to specify ( $\gamma$ ), many factors

## Gamma correction



- Want value 0 = minimum intensity
- Want value max (1 or 255) = maximum intensity
  - those 2 are easy to get
- Pick one more point
  - Midpoint should be 50%
  - Easy – show 50% black white + 50% gray
  - Adjust gamma until it looks the same
- All this happens “behind the scenes”
- Everything gets harder when we deal with color

## How to represent an image



- Now we know what intensity is,  $i = f(x,y)$ 
  - Issues with things being continuous
- Quantization – 255 levels OK for I
  - But often get less – black and white printing, ...
- Discretization / Sampling – only a finite set of points

## 2 types of image representation



- |   |  |
|---|--|
| • Raster or “image-based”   | • Vector or Geometric                                  |
| • Regular samples <ul style="list-style-type: none"><li>– Pixels</li><li>– Usually a rectangular grid<ul style="list-style-type: none"><li>• But doesn't have to be</li><li>• Hexagonal grid, ...</li></ul></li><li>– Dense</li></ul> | • Mathematical description of regions                  |
| • Does not adapt to scene   | • Exact position of points                             |
| • Store value for each regular sample   | • Mathematical descriptions of sets of points (shapes) |
| • Pixels  | • Adapts to scene                                      |
|   | • Store value for each unit                            |
|   | • Objects  |

## What is a pixel?



- Raster means regular, or uniform “grid”
- Two views of a pixel
  - A pixel is a POINT SAMPLE
    - Measurement at an infinitesimally small place
  - A pixel is finite region with constant value
    - Assumes image is collection of piecewise constant regions
- Point sample is better
  - More correct, better mathematics, can model the other

## Point Sampling Has Problems



- Miss small things
- Problem: discretization throws away information
- Don't know what happens between samples
- Sampling loses information – you cannot get back the information once its lost!

## Little squares lose differently



- Are squares better than point samples?
- Average over a little square
- But:
  - Don't know what really happened
  - Was it really constant, or was it a spike?
- Good intuition for what is coming up

## Dealing with discretization



- Sampling
  - Understand what information we are throwing away
- Reconstruction
  - Recreate as well as possible from the samples
- Re-Sampling
  - Transform the image
- Signal Processing / Image Processing
- Consider the 1D case first since its easier